user until the user enters "END" or presses the END key which signals an end to the list of components for this level. Thus, the system tests the user input in decision block **238** for the entry of "END" or the pressing of the END key. If that input is not detected, then the system waits for the next user input in decision block **232**, and when an input is received, the component is stored in the database table in function block **234** and so forth.

Once all the components have been input by the user for a given level as indicated by entering "END" or pressing the END key, the system then determines in decision block **240** if the last component in the current level has had components input by the user. If not, the next component in the current level is highlighted in the displayed tree structure, and the system loops back to function block **230** where the user is again prompted for components of this component. On the other hand, if the last component of the current level has had components inputted by the user as detected in decision block **240**, the system tests for a user input in decision block **244** to determine if components are to be entered for the next level. This is accomplished by the user pressing a Y key or an N key when prompted for the next level. If the Y key is pressed indicating that the user now wants to input the next level of components, the system loops back to block **228** to index to the next level. If on the other hand, the N key is pressed indicating that the user does not at this time wish to input the next level of components or that there is no next level of components to enter, the query process ends.

Product costing using the conceptual design tool according to the invention is illustrated by the flow chart shown in FIG. **7**. The process begins at a point where both the functional structure (FIG. **6**) and the manufacturing details (FIG. **5**) have been input. In function block **124**, the user is prompted to input product assembly time and percent contingency for estimating purposes. When this data has been input by the user, the system decomposes the structure into a parts list in function block **126**. The manner in which this is done is detailed in FIG. **8**. The system then generates the quantity and cost of the part from the manufacturing details in function block **128** and repeats this process for each part in block **130**. When all parts have been processed for quantity and cost, the cost and quantity for each part are multiplied in function blocks **132** and **134** to arrive at a series of cost figures for all parts in the product. These cost figures are summed in function block **136** together with a cost figure which is the product of assembly time multiplied by the sum of the labor and burden rates. This estimated cost figure is then output to the user by printing, for example, in function block **138**, and the process ends. The printout of the cost data includes not only the final estimated product cost but also the data from which that estimated cost is derived.

Turning now to the flow chart of FIG. **8**, the parts list in function block **126** is automatically generated as an indented bill of materials from the table in the database which was built during the query session. Again, this flow chart shows the logic of the automatic generation of an indented bill of materials, and any programmer skilled in the art with an understanding of database systems, such as the IBM DB2 database, can write code to implement the invention from the logic of the flow chart. The process begins in FIG. **8** by setting $l=1$ and $i=0$ in block **246**, where is the component level as before and i is the indentation of the bill of materials. Next, item **1** of level l is accessed in function block **248**. In the example given, this item is the product name "LAWNMOWER". Item 1 is then printed in function block **250**, and l and i are then indexed by adding **1** to each. A test is then made in decision block **254** to determine if any level l is left in the tree. If so, the system accesses the next left-most item in the tree of the current level in function block **256**. The accessed item is then printed in function block **258** with indentation i. A search is then made of the database in function block **260** for antecendents. If any are found in decision block **262**, the system loops back to block **252** where the level and indentation are indexed by one. Otherwise, a test is made in decision block **264** to determine if the last item of the current level has been connected. If so, the level and the indentation are indexed backward in block **266** by subtracting one from each. The process then returns to decision block **254** to continue the process of accessing and printing items in order. When the test in decision block **254** becomes negative, that is there are no levels l left in the tree structure, the level and indentation are again indexed backward by subtracting one in block **268**. A test is then made in decision block **270** to determine if the indentation i is less than or equal to zero. If not, the process loops back to decision block **254**; otherwise, the indented bill of materials is complete and the process ends.

From the foregoing, it will be appreciated that the conceptual design tool according to the invention provides an easy to use system which implements a top-down functional approach to hardware product design. The system encourages early manufacturing involvement to develop the information needed to aid and improve the total design and manufacturing effort needed to produce the final product.

While the invention has been described in terms of a preferred embodiment, those skilled in the art will recognize that the invention can be practiced with modification and alteration within the spirit and scope of the appended claims.

Having thus described our invention, what we claim as new and desire to secure by letters patent is as follows:

1. A conceptual design tool method for implementation on a computer, said conceptual design tool method providing for the early manufacturing involvement information needed to aid and improve the total design planning and manufacturing effort to product a final product, comprising the steps of:

prompting a user on a functional product structure;

capturing product structure data input by the user in a database by entering components for each level of component detail in a table in the database;

generating a hierarchical structure of the product structure and displaying said hierarchical structure to the user as the product structure data is captured in the database;

prompting the user to input manufacturing details for each item in the functional product structure by selecting an item in the structure and displaying a screen for input of manufacturing details;

providing the user with an option to enter default manufacturing information of each item;

accessing a database to insert default manufacturing information in said screen from said database for similar items;

providing the user with an option to override default manufacturing information inserted in said screen;